

# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Assistant Commissioner for Patents  
Washington, D.C. 20231

Atty. Dkt.: 2018-477

Date: December 18, 2001

Sir:

Attached for filing is the patent application of:

Inventor: NISHIMURA

Entitled: PROCESSOR UNIT FOR EXECUTING EVENT PROCESS IN  
REAL TIME IN RESPONSE TO OCCURRENCE OF EVENT

and including attachments as noted below:

☒ Newly executed Declaration, ☐ Copy of Declaration from prior application, ☒ Abstract  
22 pages of specification and claims (including 12 numbered claims), and  
3 sheets of accompanying drawing/s.

☒ Record the attached assignment and return to the undersigned.

☐ Attached is a Power of Attorney.

☒ Priority is hereby claimed under 35 U.S.C. § 119 based on the following foreign applications:

**Application Number**

**Country**

**Day/Month/Year Filed**

2000-387120

JAPAN

20 December 2000

, respectively, the entire content of which is hereby incorporated by reference in this application..

☒ Certified copy(ies) of foreign application(s) is/are attached.

☐ Certified copy(ies) filed on \_\_\_\_\_ in prior appln. no. \_\_\_\_\_ filed \_\_\_\_\_

☐ Please amend the specification by inserting the following paragraph before the first line: --This application claims the benefit of Provisional Application No. \_\_\_\_\_, filed \_\_\_\_\_, the entire content of which is hereby incorporated by reference in this application.--

☐ Please amend the specification by inserting the following paragraph before the first line: --This application is a ☐ continuation/☐ division/☐ continuation-in-part of Application No. \_\_\_\_\_, filed \_\_\_\_\_, the entire content of which is hereby incorporated by reference in this application.--

☐ Petition filed in prior application to extend its life to insure co-pendency.

☐ The prior application is assigned to \_\_\_\_\_

☐ It is hereby requested that the Examiner consider the art cited in the parent application by applicant and/or the Examiner for the reasons stated therein. A listing of that art is attached.

☐ Applicant claims "small entity" status. ☐ "Small entity" statement attached.

☐ Please enter the attached and/or below preliminary amendment prior to calculation of filing fee:

☒ Also attached: ☒ Information Disclosure Statement; ☐ Non-Publication Request; ☐ Nucleotide and/or Amino Acid Sequence Submission; ☐ Statement deleting Inventor(s) named in prior application; ☒ Other: PTO 1449 W/References

## FILING FEE IS BASED ON CLAIMS AS FILED LESS ANY HEREWITH CANCELED

Basic Filing Fee				\$	740.00
Total effective claims	12	- 20 (at least 20) =	0	x \$	18.00
				\$	0.00
Independent claims	4	- 3 (at least 3) =	1	x \$	84.00
				\$	84.00
If any proper multiple dependent claims now added for first time, add \$280.00 (ignore improper)				\$	0.00
					<b>SUBTOTAL</b>
				\$	824.00
If "small entity," then enter half (1/2) of subtotal and subtract				-\$	(0.00)
					<b>SECOND SUBTOTAL</b>
				\$	824.00
Assignment Recording Fee (\$40.00)				\$	40.00
					<b>TOTAL FEE ENCLOSED</b>
				\$	864.00

Any future submission requiring an extension of time is hereby stated to include a petition for such time extension.

The Commissioner is hereby authorized to charge any deficiency, or credit any overpayment, in the fee(s) filed, or asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our **Account No. 14-1140**.  
A duplicate copy of this sheet is attached.

1100 North Glebe Road, 8<sup>th</sup> Floor  
Arlington, Virginia 22201-4714  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100  
LSN:pdcc

**NIXON & VANDERHYE P.C.**

By Atty: Larry S. Nixon, Reg. No. 25,640

Signature: \_\_\_\_\_

*Larry S. Nixon*

#3  
11017 U.S. PTO  
10/020437  
12/18/01

J1017 U.S. PTO  
10/020437  
12/18/01

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年12月20日

出 願 番 号

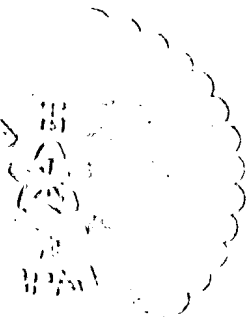
Application Number:

特願2000-387120

出 願 人

Applicant(s):

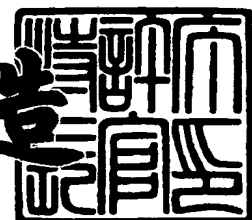
株式会社デンソー



2001年 9月25日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3087693

【書類名】 特許願

【整理番号】 PNID3609

【提出日】 平成12年12月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町 1 丁目 1 番地 株式会社デンソー内

    【氏名】 西村 忠治

【特許出願人】

    【識別番号】 000004260

    【氏名又は名称】 株式会社デンソー

【代理人】

    【識別番号】 100082500

    【弁理士】

    【氏名又は名称】 足立 勉

    【電話番号】 052-231-7835

【手数料の表示】

    【予納台帳番号】 007102

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9004766

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 処理実行装置及び記録媒体

【特許請求の範囲】

【請求項 1】

所定のイベントの発生を検知し、当該イベントに応じたイベント対応処理を実行するタスクスケジューリングプログラムを備えた処理実行装置において、

前記タスクスケジューリングプログラムは、

前記イベント対応処理を実現するためのタスクと、

前記イベントの発生を検知して、該当するタスクの起床を要求する起床要求プログラムと、

前記起床要求プログラムからの要求に対し、設定された優先度に基づいて前記タスクを起床するリアルタイムオペレーティングシステムとを有しており、

前記タスクに複数のイベント対応処理が記述されている場合には、前記起床要求プログラムが、該当するイベント対応処理を特定するための識別情報をメモリ領域に格納し、一方、前記タスクは、当該メモリ領域に格納された前記識別情報を読み出して、該当するイベント対応処理を実現することを前提として、

前記イベントの少なくとも一部である特定イベントに対しては、当該特定イベント毎に、当該特定イベントに対応する処理のみを実現するための専用タスクを用意することによって、前記識別情報による前記イベント対応処理の特定を不要にしたこと

を特徴とする処理実行装置。

【請求項 2】

請求項 1 に記載の処理実行装置において、

前記特定イベントは、時間に同期して発生する周期的なイベントであることを特徴とする処理実行装置。

【請求項 3】

請求項 1 又は 2 に記載の処理実行装置において、

前記処理実行装置は、所定の制御対象を制御する制御装置として実現されており、

前記特定イベントは、前記制御対象の運転状態に同期して発生する周期的なイベントであること

を特徴とする処理実行装置。

【請求項 4】

請求項 1 ～ 3 のいずれかに記載の処理実行装置において、

前記特定イベントは、本装置を含むシステムの変更があっても追加あるいは削除される可能性が相対的に小さなイベントであること

を特徴とする処理実行装置。

【請求項 5】

請求項 1 ～ 4 のいずれかに記載の処理実行装置において、

前記特定イベントは、前記起床要求プログラムから前記タスクへのデータの受け渡しを必要としないイベントであること

を特徴とする処理実行装置。

【請求項 6】

請求項 1 ～ 5 のいずれかに記載の処理実行装置において、

前記特定イベント以外のイベントに対応するイベント対応処理は、前記優先度毎に用意された共有タスクで実現されること

を特徴とする処理実行装置。

【請求項 7】

請求項 1 ～ 6 のいずれかに記載のタスクスケジューリングプログラムを記録したこと

を特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、リアルタイムオペレーティングシステム（以下「RTOS」という。）を用いたタスク切り換え技術に関する。

【0002】

【従来の技術】

例えば車両に搭載される電子制御装置（以下「ECU」という。）に用いられる制御プログラムでは、制御応答性や安全性の確保等から所定の処理をリアルタイムに実行する必要がある。そのため、優先度の設定されたタスクの単位で処理プログラムを記述し、RTOSによるタスクの切り換えによって、リアルタイム性を確保している。

#### 【0003】

より具体的には、イベントの発生を検知して起床要求プログラムが該当タスクの起床要求を行うと、RTOSがタスクを起床し、起床されたタスクにて、そのイベントに応じた処理（以下「イベント対応処理」という。）が実行されるようになっている。なお、タスクにはそれぞれ優先度が設定されている。これはタスク間に実行の優先順位が存在するからであり、リアルタイム性の高い処理ほど、優先度の高いタスクとして実現される。したがって、RTOSは、起床要求のあったタスクが起床されておらず、かつ、既に起床されているタスクがあれば、そのタスクよりも起床要求のあったタスクの優先度が高い場合に、その起床要求のあったタスクを起床する。

#### 【0004】

従来、このRTOSによって起床されるタスクは、上述した優先度毎に、設けられていた。つまり、優先度が同じである複数のイベント対応処理が、1つのタスクに属するプログラム構造になっていた。これについて説明する。

RTOSによって起床されるタスクの優先度が3段階に設定されていれば、図5に示すように、高優先度のAタスク、中優先度のBタスク、低優先度のCタスクという具合に3つのタスクが存在する。一例として各タスクで実行されるイベント対応処理を挙げると、Aタスクで実行される処理は、1ms毎に実行される1ms処理、クランク軸の30°CA毎に実行される30°CA処理などである。また、Bタスクで実行される処理は、4ms毎に実行される4ms処理、8ms毎に実行される8ms処理、所定気筒のピストンが上死点（TDC）に来た時点で実行されるTdc処理、エンジンストール時に実行されるエンスト処理、自己診断を行うための故障検出処理などである。さらに、Cタスクで実行される処理は16ms毎に実行される16ms処理、32ms毎に実行される32ms処

理などである。

【0005】

このようにタスクを優先度毎に設けている理由は、RTOSがタスクの切り換えを優先度で行うからであり、また、タスクの総数がそれほど多くならず、RTOSによる管理情報が比較的少なくなるためである。管理情報が減少すれば、RAMといったメモリの記憶容量を削減できる点で有利である。

【0006】

【発明が解決しようとする課題】

しかしながら、上述したプログラム構造では、1つのタスク内に複数のイベント対応処理が存在するため、要求されたイベント対応処理を各タスクが特定して実行する必要が生じる。

【0007】

具体的には、起床要求プログラムが、イベントの発生を検知すると、RTOSに対し該当タスクの起床要求を行うと共に、実行すべきイベント対応処理を特定するための識別情報(ID)を、待ち行列であるキュー(メモリ領域)に格納する。一方、タスクは、RTOSによって起床されると、キューからIDを読み出し、要求されたイベント対応処理を特定して実行する。

【0008】

IDのキューへの記憶及びキューからの取得は、FIFOと呼ばれる方式で行われる。FIFOは、ある場所に格納したデータを、古く格納した順に取り出すようにする方式である。これは、様々なイベントが連続して発生する可能性を考慮したものである。

【0009】

以上のように、従来のプログラム構造では、タスクが優先度毎に設けられていたため、イベントが発生する度に必ず、IDの記憶及び取得(以下適宜「キューイング」という。)が必要になっていた。このキューイングでは、キューにデータがあるか否かのチェックや、キューが一杯になっているか否かのチェックも必要である。そのため、従来のプログラム構造では、各種のイベントが連続して発生する状況下において、CPUの処理負荷が大きなものになっていた。

【0010】

そしてCPUの処理負荷が大きくなると、相対的に優先度の低いタスクの起床が遅れ、結果的にそのタスク内のイベント対応処理の実行が遅れることになる。また、CPUの負荷が100パーセントに達すると、相対的に優先度の低いイベント対応処理は間引かれて、実行されなくなってしまう。その結果、応答性が悪化するなど、制御性能への悪影響が生じる。

【0011】

本発明は、上述した問題点を解決するためになされたものであり、上述したイベント対応処理をRTOSに管理されるタスクで実現するにあたり、CPUの処理負荷を低減させることを目的とする。

【0012】

【課題を解決するための手段及び発明の効果】

上述した目的を達成するためになされた請求項1に記載の処理実行装置は、所定のイベントの発生を検知し、当該イベントに応じたイベント対応処理を実行するタスクスケジューリングプログラムを備えている。

【0013】

なお、本明細書では、「プログラムが・・・する」といったプログラムを主体とした表現を適宜用いるが、詳しくは、処理実行装置のCPUがプログラムに基づく処理を実行することによって各機能が実現されることは言うまでもない。

このタスクスケジューリングプログラムは、イベント対応処理を実現するためのタスクと、起床要求プログラムと、RTOSとを有している。そして、起床要求プログラムがイベントの発生を検知して該当するタスクの起床を要求すると、RTOSは、起床要求プログラムからの要求に応じ、設定された優先度に基づいてタスクを起床する。

【0014】

このとき、タスクに複数のイベント対応処理が記述されている場合には、起床要求プログラムが、該当するイベント対応処理を特定するための識別情報をメモリ領域に格納し、一方、タスクは、当該メモリ領域に格納された識別情報を読み出して、該当するイベント対応処理を実現する。



## 【0015】

このような基本構成を前提として、本発明では、特定イベントに対しては、当該特定イベント毎に、当該特定イベントに対応するイベント対応処理のみを実現するための専用タスクを用意することによって、識別情報によるイベント対応処理の特定を不要にした。

## 【0016】

つまり、本発明の技術思想は、イベントとタスクとを1対1に対応させることによって、従来技術で説明したような識別情報のキューイングを不要にし、CPUの処理負荷を軽減しようとするものである。

なお、特定イベントは、発生するイベントの中の少なくとも一部とすることが考えられる。全てのイベントとタスクとを1対1に対応させてもよいが、その場合には、タスク数が膨大になり、RTOSの管理情報が多くなってしまうからである。また、イベントによって識別情報のキューイングが連続的に生じることによってCPUの処理負荷が増大するという事実を考えると、単発的に発生するイベントに対して専用タスクを割り当てる必要性は小さい。

## 【0017】

そこで、請求項2に示すように、特定イベントは、時間に同期して発生する周期的なイベントとすることが考えられる。例えば所定時間間隔で繰り返し発生するイベントを特定イベントにするという具合である。また、請求項3に示すように、処理実行装置が所定の制御対象を制御する制御装置として実現されている場合、特定イベントは、制御対象の運転状態に同期して発生する周期的なイベントとしてもよい。ここで制御対象の一例としては、車両のエンジンが考えられる。制御対象が車両のエンジンである場合、例えば、エンジンのクランク軸の回転に同期した回転同期イベントを特定イベントにするという具合である。

## 【0018】

このように周期的に繰り返されるイベントを特定イベントとして、この特定イベントに専用タスクを対応させ、識別情報のキューイングを不要にすれば、CPUの処理負荷を大幅に削減することができる。

また、例えば制御対象が車両のエンジンなどである場合、モデルチェンジのタ

イミングや設計変更などで、追加／削除されるイベントもある。このようなイベントの追加／削除があると、該当するイベント対応処理の変更が必要になる。しかし、専用タスクとしてそのイベント対応処理を実現する場合、タスク構成自体を変更する必要が生じて、変更作業に手間がかかる。

#### 【 0 0 1 9 】

そこで請求項 4 に示すように、本装置を含むシステムの変更があっても追加あるいは削除される可能性が相対的に小さなイベントを、特定イベントとすることが望ましい。このようにすれば結果的に、専用タスクを変更する必要が少なくなり、タスク構成自体の変更が少なくなって、イベントの追加／削除による変更作業が容易になるからである。

#### 【 0 0 2 0 】

さらにまた、上述したイベントはイベント対応処理のトリガとなるものであるが、このようなトリガに加え処理対象のデータを含むメッセージ（イベント＋データ）が存在する。この場合には、このデータの受け渡しにおいてキューイングが必要になるため、専用タスクとしてもデータのキューイングは発生することになる。したがって、請求項 5 に示すように、起床要求プログラムからタスクへのデータの受け渡しを必要としないイベントを、特定イベントとすることが考えられる。

#### 【 0 0 2 1 】

なお、車両のエンジンを制御対象とする場合、周期的なイベントは、一般的に追加／削除される可能性が少なく、また、データの受け渡しを要しない場合が多い。

ところで、請求項 6 に示すように、上述してきた発明において、特定イベント以外のイベントに対応するイベント対応処理は、従来技術と同様に、優先度毎に用意された共有タスクで実現することが考えられる。タスクの切り換えを優先度毎に行う R T O S の管理手法から見て適切だからであり、また、タスクの総数を抑えることができ、R T O S による管理情報を比較的少なくできるためである。管理情報が減少すれば、R A M といったメモリの記憶容量を削減できる。つまり、C P U の処理負荷低減に有効なイベントに対する処理は専用タスクとして実現

し、一方、そうでないイベントに対する処理は優先度毎の共有タスクとして実現することによって、共有タスクとして実現した場合のメリットも享受できるのである。

#### 【 0 0 2 2 】

なお、上述したタスクスケジューリングプログラムの場合、例えば、FD、MO、CD-ROM、DVD、ハードディスク等のコンピュータ読み取り可能な記録媒体に記録し、必要に応じてコンピュータシステムにロードして起動することにより用いることができる。この他、ROMやバックアップRAMをコンピュータ読み取り可能な記録媒体としてプログラムを記録しておき、このROMあるいはバックアップRAMをコンピュータシステムに組み込んで用いてもよい。

#### 【 0 0 2 3 】

##### 【発明の実施の形態】

以下、本発明を具体化した一実施例を図面を参照して説明する。

図1は、本発明の「処理実行装置」を具体化したエンジン制御装置（以下「ECU」という。）1の構成を表すブロック図である。ECU1は、車両に搭載された内燃機関型エンジンの制御を行う。

#### 【 0 0 2 4 】

ECU1は、エンジンのクランク軸が所定角度回転する毎にパルス状の信号を出力する回転角センサ、エンジンの特定の気筒のピストンが所定位置（例えば上死点：TDC）にくる度にパルス状の信号を出力する基準位置センサ、エンジンの冷却水の温度を検出する水温センサ、及び酸素濃度を計測する酸素濃度センサ等、エンジンの運転状態を検出する様々なセンサ30からの信号を入力して波形整形やA/D変換を行う入力回路21と、入力回路21からのセンサ信号に基づき、エンジンを制御するための様々な処理を実行するマイコン10と、マイコン10からの制御データに応じて、エンジンに取付けられたインジェクタ（燃料噴射装置）及びイグナイタ（点火装置）等のアクチュエータ40を駆動する出力回路22とを、備えている。

#### 【 0 0 2 5 】

そして、マイコン10には、プログラムを実行する周知のCPU11と、CP

U11によって実行されるプログラムを記憶するROM12と、CPU11による演算結果等を記憶するためのRAM13と、入力回路21及び出力回路22との間で信号をやり取りするためのI/O14と、各種レジスタやフリーランカウンタ等（図示省略）とが備えられている。

【0026】

このように構成されたECU1は、各種センサ30から入力回路21を介して入力される信号に基づき、出力回路22に接続されたアクチュエータ40を駆動する、エンジン制御処理を行う。

次に図2に基づき、ROM12に記憶される「タスクスケジューリングプログラム」としてのエンジン制御プログラムの構成を説明する。エンジン制御プログラムは、少なくとも、起床要求プログラム12a、RTOS12b、及び、タスク12c、12dから構成されている。

【0027】

起床要求プログラム12aの実行により、イベントの発生が検知されると、起床要求プログラム12a中の起床要求処理が実行される。この起床要求処理では、検知されたイベントに対応するタスク12c、12dを特定し、RTOS12bに対して、タスク12c、12dの起床を要求する。なお、イベントの発生は、上述したセンサ群30からの信号に基づいて検知することが考えられる。また、マイコン10内部のフリーランカウンタによって検知することが考えられる。さらに、車内LAN等のネットワーク情報によって検知することが考えられる。

【0028】

RTOS12bは、タスク12c、12dの起床要求があると、起床要求のあったタスク12c、12dが実行中でない場合で、かつ、既に起床されているタスク12c、12dの優先度が起床要求のあったタスク12c、12dの優先度よりも低い場合に、起床要求のあったタスク12c、12dを起床する。これによって優先度に基づくタスク12c、12dの起床（切り換え）が行われ、リアルタイム性の高いタスク12c、12dが優先的に実行される。

【0029】

特に本実施例では、タスク12c、12dの構成を工夫しており、A、B、C

の3つの共有タスク12cと、D、E、F、G、H、I、Jの7つの専用タスク12dを備える構成とした。

本実施例では、特定のイベントに対しては、そのイベントに対応するイベント対応処理のみを実現するD～Jの専用タスク12dを用意した。ここで特定のイベントとは、制御対象であるエンジン回転に同期して発生するイベント及び、時間に同期して発生する所定時間間隔のイベントである。すなわち、図2では、図5に示した処理の中の、1ms処理、4ms処理、8ms処理、16ms処理、32ms処理、30°CA処理、及びTd処理が、D～Jのそれぞれの専用タスク12dに記述されている。

#### 【0030】

一方、特定イベント以外のイベントに対応する処理については、優先度毎に用意されたA～Cの共有タスク12cに記述した。図2では、図5に示した処理の中の、エンスト処理、故障検出処理が、中優先度のB共有タスクに記述されている。

#### 【0031】

次に、起床要求処理及びタスク処理が共有タスク12cと専用タスク12dとでどのように異なるかを、図3及び図4のフローチャートを用いて説明する。ここでは最初に共有タスク12cの場合を説明し、続けて専用タスク12dの場合を説明する。

#### 【0032】

図3(a)は、共有タスク12cの起床を要求する起床要求処理を示すフローチャートである。

まず最初のステップ(以下、ステップを単に記号Sで示す。)100において、タスク12cを選択する。この処理は、発生したイベントに対応する処理を特定するためのIDに基づいて行われる。

#### 【0033】

続くS110では、IDをキューに記憶する。そして必要であれば、IDと共にデータもキューに記憶する。このデータはイベントに関連するものであり、この処理は、共有タスク12cに受け渡す必要がある場合に行われる。

続く S 1 2 0 では、S 1 0 0 にて選択した共有タスク 1 2 c の起床を、R T O S 1 2 b に対して要求する。そして、その後、本起床要求処理を終了する。

【0 0 3 4】

このような起床要求処理によって、上述したように R T O S 1 2 b が該当する共有タスク 1 2 c を起床する。

図 3 ( b ) は、R T O S 1 2 b に起床された共有タスク 1 2 c におけるタスク処理を示すフローチャートである。

【0 0 3 5】

まず最初の S 2 0 0 では、キューに記憶された I D を読み出して取得する。そして、上述したようにイベントに関連するデータがキューに記憶されている場合には、そのデータも I D と共に読み出して取得する。

次の S 2 1 0 では、I D に基づき、実行対象のイベント対応処理を特定する。

【0 0 3 6】

そして続く S 2 2 0 では I D より特定したイベント対応処理を実行し、その後、本タスク処理を終了する。

このように共有タスク 1 2 c は優先度毎に用意されており、1 つの共有タスク 1 2 c に複数のイベント対応処理が記述されている。そのため、共有タスク 1 2 c に係る起床要求処理（図 3 ( a ) 参照）では、イベント対応処理を特定するための I D をキューに記憶する必要がある（S 1 1 0）。一方、タスク処理（図 3 ( b ) 参照）では、I D をキューから取得し（S 2 0 0）、さらに、I D に基づいてイベント対応処理を特定する必要がある（S 2 1 0）。

【0 0 3 7】

これに対し、専用タスク 1 2 d に係る起床要求処理及びタスク処理を次に説明する。

図 4 ( a ) は、専用タスク 1 2 d の起床を要求する起床要求処理を示すフローチャートである。

【0 0 3 8】

まず最初の S 3 0 0 において、専用タスク 1 2 d を選択する。この処理は、共有タスクに係る処理（図 3 ( a ) 中の S 1 0 0）と同様のものであり、発生した

イベントに対応する処理を特定するためのIDに基づいて行われる。

そして専用タスク12dは特定イベントに1対1に対応させて用意され、そのイベントに対応するイベント対応処理のみが記述されている。したがって、S300におけるタスク選択後、次のS310では専用タスク12dの起床要求を行い、その後、本起床要求処理を終了する。

#### 【0039】

つまり、共有タスク12cの場合と異なり、図3(a)の起床要求処理に示したS110に相当する処理が不要になる。

起床要求処理によって、上述したようにRTOS12bが該当する専用タスク12dを起床する。

#### 【0040】

図4(b)は、RTOS12bに起床された専用タスク12dにおけるタスク処理を示すフローチャートである。

専用タスク12dには、ある特定イベントに対応するイベント対応処理のみが記述されているため、S400にてそのイベント対応処理を実行し、その後、本タスク処理を終了する。

#### 【0041】

つまり、図3(b)中のS200及びS210に相当する処理が不要になる。

以上のようにタスク12c、12dを構成したことによる効果を、次に説明する。

上述したようなIDの記憶(図3(a)中のS110)及び取得(図3(b)中のS200)では、キューが空であるか否かのチェックやキューが情報で一杯になっているか否かのチェックも必要になるため、発生する全てのイベントに対して、これらの処理が発生することによって、CPUの処理負荷が大きくなっていた。特に、エンジン回転に同期したイベントや時間に同期したイベントといった周期的なイベント発生によるIDのキューイングがCPUの処理負荷を増大させていた。

#### 【0042】

本実施例では、このような周期的なイベントを特定イベントとし、これらのイ

イベントに関しては、イベント毎に用意される専用タスク 1 2 d にて、イベント対応処理を実現する。これによって、上述したような、ID の記憶（図 3（a）中の S 1 1 0）及び取得（図 3（b）中の S 2 0 0）が不要になる。これによって、周期的なイベントによる ID のキューイングがなくなるため、CPU の処理負荷を大幅に低減することができる。

#### 【0043】

また、本実施例では、周期的なイベント、すなわち、CPU の処理負荷低減に有効なイベントに対する処理は専用タスク 1 2 d として実現し、一方、そうでないイベントに対する処理は優先度毎の共有タスク 1 2 c として実現した。

例えばエンジンストールや故障検出といった単発的なイベントにおいては、ID のキューイングが発生しても、その処理時間がそれほど問題にならないからである。そして、このようなイベント対応処理を優先度毎に用意された共有タスク 1 2 c にて実現すれば、優先度に基づいてタスク 1 2 c, 1 2 d を起床する RTOS の管理手法から見て適切であり、また、タスクの総数を抑えることができ、RTOS による管理情報を比較的少なくできる。管理情報が減少すれば、RAM といったメモリの記憶容量を削減できる。つまり、本実施例によれば、CPU 1 の処理負荷を低減すると共に、共有タスク 1 2 c として実現した場合のメリットも享受できるのである。

#### 【0044】

以上、本発明はこのような実施例に何等限定されるものではなく、本発明の主旨を逸脱しない範囲において種々なる形態で実施し得る。

（イ）例えば、上記実施例では、特定イベントを、エンジンの回転に同期して発生するイベント及び時間に同期して発生するイベントとしていたが、ECU 1 を含むエンジン制御システムの変更を考慮して特定イベントを決定することも考えられる。すなわち、システムの変更時に追加あるいは削除される可能性が相対的に小さなイベントを、特定イベントとする。このようにすればシステムの変更があっても、専用タスク 1 2 d を変更する必要が少なくなり、タスク構成自体の変更が少なくなつて、イベントの追加／削除に伴うプログラムの変更作業が容易になる。



【0045】

(ロ) 上述したイベントはイベント対応処理のトリガとなるものであるが、このようなトリガと共に処理対象のデータを含むメッセージ（イベント＋データ）が存在する。この場合には、このデータの受け渡しにおいてキューイングが必要になるため、専用タスク 1 2 d としてイベント対応処理を実現しても、図 3（a）中の S 1 1 0 におけるデータの格納処理、及び図 3（b）中の S 2 0 0 におけるデータの取得処理が必要になり、I D のキューイングを不要にしても CPU の処理負荷がほとんど変わらない。したがって、起床要求プログラム 1 2 a からタスク 1 2 c、1 2 d へのデータの受け渡しを必要としないイベントを、特定イベントとすることが好ましい。

【0046】

(ハ) 上記実施例は、車両のエンジンを制御する ECU 1 として本発明を具現化したものであった。これに対し、RTOS によるタスクスケジューリングを行う全ての処理実行装置に本発明が適用できることは言うまでもない。

【図面の簡単な説明】

【図 1】 実施例のエンジン制御装置の構成を表すブロック図である。

【図 2】 エンジン制御プログラムの構造を示す説明図である。

【図 3】 共有タスクに係る起床要求処理及びタスク処理を示すフローチャートである。

【図 4】 専用タスクに係る起床要求処理及びタスク処理を示すフローチャートである。

【図 5】 従来のタスクスケジューリングプログラムの構造を示す説明図である。

【符号の説明】

1 … ECU	1 0 … マイコン
1 1 … CPU	1 2 … ROM
1 2 a … 起床要求プログラム	1 2 b … RTOS
1 2 c … 共有タスク	1 2 d … 専用タスク
1 3 … RAM	1 4 … I/O

2 1 … 入力回路

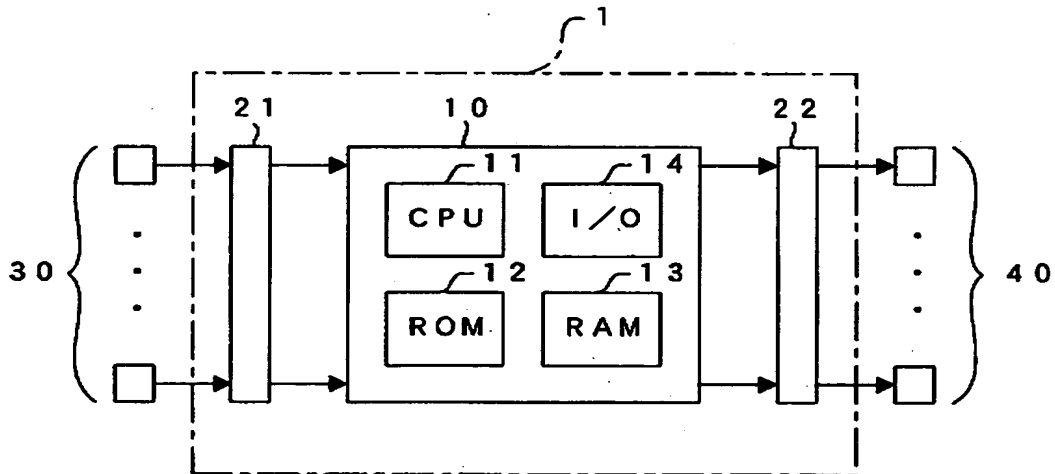
2 2 … 出力回路

3 0 … センサ

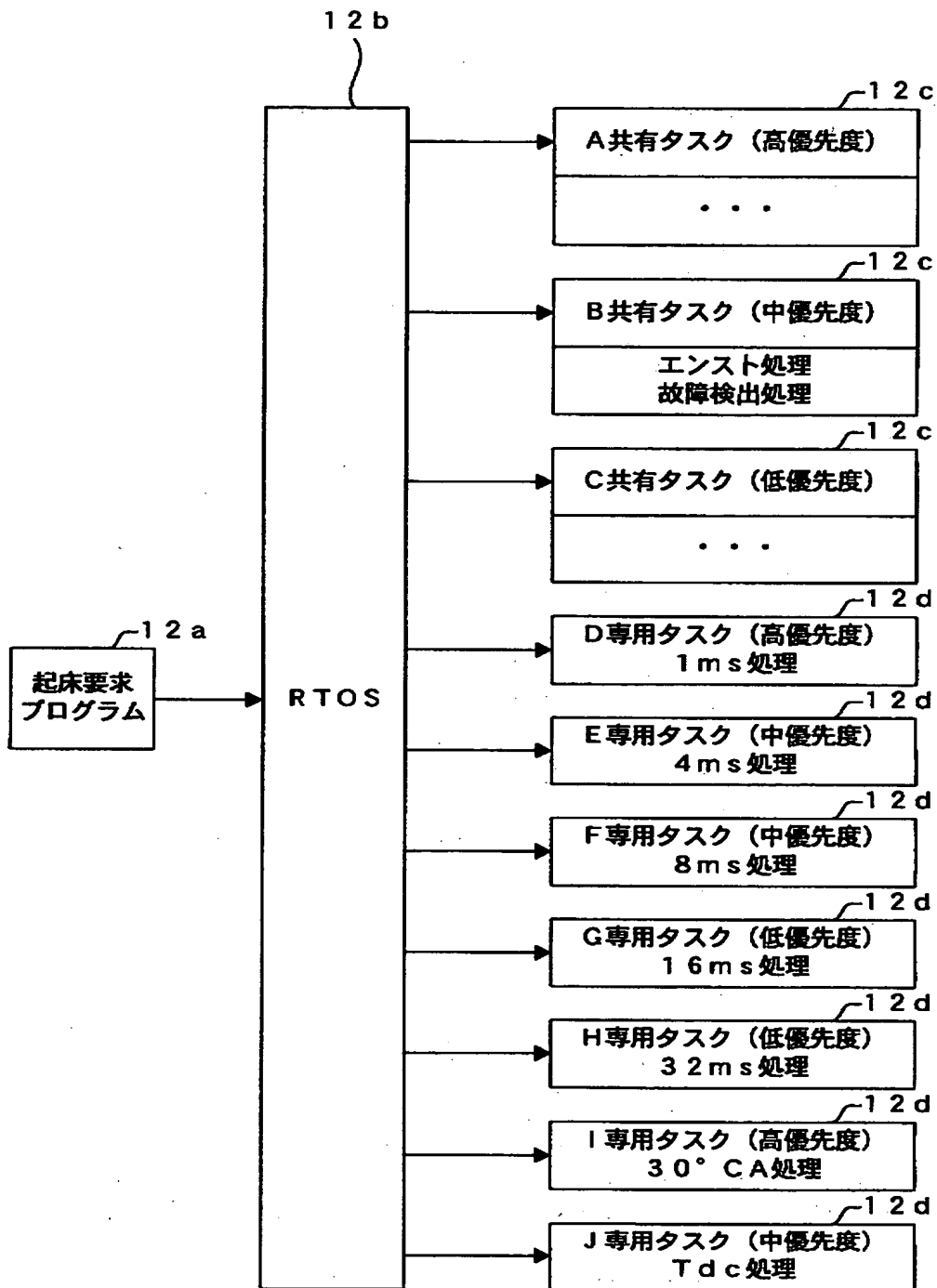
4 0 … アクチュエータ

【書類名】 図面

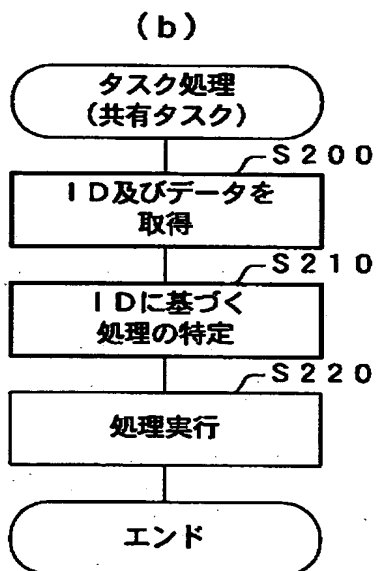
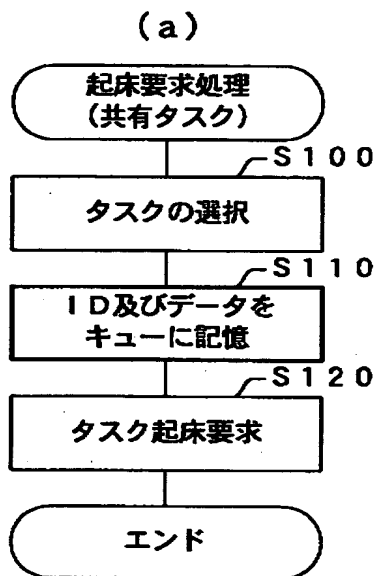
【図 1】



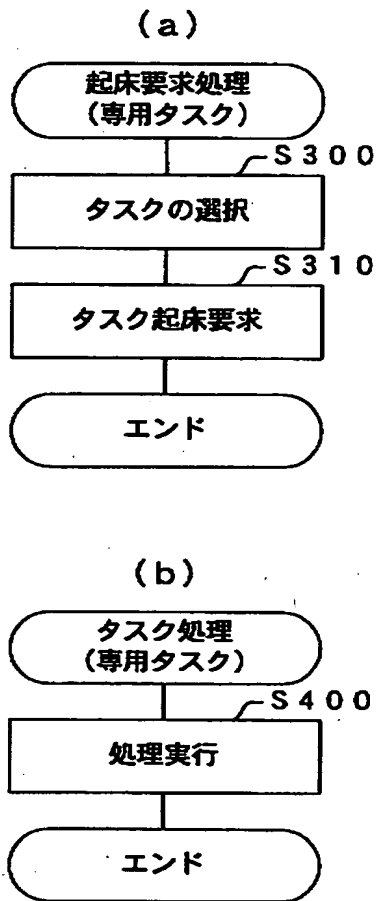
【図 2】



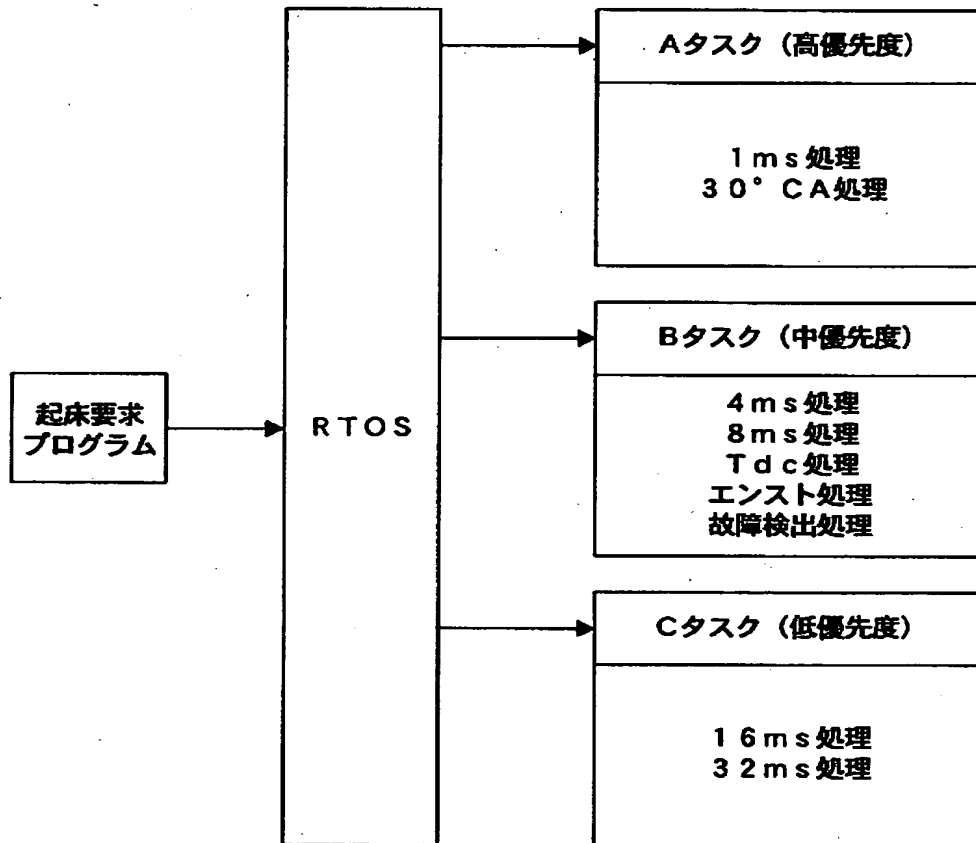
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 イベントに応じたイベント対応処理をR T O S（リアルタイムオペレーティングシステム）により管理されるタスクで実現するにあたり、C P Uの処理負荷を低減させる。

【解決手段】 起床要求プログラム1 2 aからの起床要求に基づき、タスク1 2 c, 1 2 dを起床するR T O S 1 2 bを備えるエンジン制御のプログラム構造において、制御対象であるエンジン回転に同期して発生するイベント及び、時間に同期して発生する所定時間間隔のイベントに対応する処理、すなわち、1 m s 処理、4 m s 処理、8 m s 処理、1 6 m s 処理、3 2 m s 処理、3 0° C A処理、及びT d c処理を、D～Jの専用タスク1 2 dにて実現し、処理特定のためのI Dのキューイング動作を不要にする。

【選択図】 図 2



特2000-387120

出 願 人 履 歴 情 報

識別番号 [000004260]

1. 変更年月日 1996年10月 8日  
[変更理由] 名称変更  
住 所 愛知県刈谷市昭和町1丁目1番地  
氏 名 株式会社デンソー